TEXAS INSTRUMENTS

TMS320C8x Emulator

Installation Guide

1995

Microprocessor Development Systems

TMS320C8x Emulator Installation Guide

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

WARNING

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

TRADEMARKS

MS-DOS is a registered trademark of Microsoft Corporation.

Pentium is a trademark of Intel Corporation.

Windows NT is a trademark of Microsoft Corporation.

Contents

Figures

Tables

Chapter 1

Installing the Emulator and C Source Debugger

This chapter helps you install the 'C8x emulator board and the C source debugger on a 32-bit x86-based or Pentium™ PC running Windows NT™ Workstation version 3.5 or later. When you complete the installation, turn to the *TMS320C80 (MVP) C Source Debugger User's Guide.*

Topic **Page**

1.1 What You'll Need

The following checklists detail items that are shipped with the 'C8x C source debugger and emulator and additional items you'll need to use these tools.

Hardware checklist

□ **host** — A 32-bit x86-based or Pentium™ PC with an ISA/EISA bus

□ **memory** — Minimum of 16 megabytes of RAM plus 32 megabytes of hard-disk space for swap files

□ **display** — Monochrome or color (color recommended)

□ **slot** — One 16-bit slot

□ **emulator board power requirements** — Approximately 1 ampere @ 5 volts (5 watts)

□ **target system** — A board with a 'C8x device

□ **connector to target system** — 14-pin connector (two rows of seven pins)—for more information about this connector, refer to the *JTAG/MPSD Emulation Technical Reference.*

□ **required hardware** — A CD-ROM drive

□ A Microsoft-compatible mouse

□ **optional hardware** — An EGA- or VGA-compatible graphics display card and a large monitor. The debugger has two options that allow you to change the overall size of the debugger display. If you have an EGA- or VGA-compatible graphics card, you can take advantage of some of these larger screen sizes. These larger screen sizes are most effective when used with a large (17" or 19") monitor. (To use a larger screen size, you must invoke the debugger with an appropriate option. For more information about options, refer to the *Debugger Options* section in the *Overview of a Code Development and Debugging System* chapter of the *TMS320C80 (MVP) C Source Debugger User's Guide.*)

Software checklist

□ **operating system** — Windows NT Workstation (version 3.5 or later)

□ **software tools** — TMS320C8x assembler, linker, and C compiler

□ **CD-ROMs** — *TMS320C80 (MVP) Online Reference*

□ *TMS320C8x Emulator*

□ **required files** † *mpemu.exe* is the executable file that invokes the MP version of the C source debugger for the emulator.

□ † *ppemu.exe* is the executable file that invokes the PP version of the C source debugger for the emulator.

□ † *emurst.exe* resets the 'C8x emulator.

□ † *pdm.exe* is the executable file for the parallel debug manager environment.

□ † *board.dat* describes your target board to your debugger in terms of what devices are in the emulation scan path.

□ † *genport.sys* is the Windows NT emulator device driver.

□ † *regdrv.exe* is an installation program that registers the emulator driver into the Windows NT registry database.

□ **optional files** † *composer.exe* is a utility that allows you to convert your text board configuration file (*board.cfg*) into a format the debugger can read (*board.dat*).

□ † *board.cfg* is a text file that describes your target board in terms of what devices are on the emulation scan path.

□ † *init.cmd* is a general-purpose batch file that contains debugger commands. The version of this file that's shipped with the debugger defines a 'C8x memory map. If this file isn't present when you first invoke the debugger and you don't use the –t option, all memory is invalid initially. When you first start using the debugger, this memory map should be sufficient for your needs. Later, you may want to define your own memory map. For information about setting up your own memory map, refer to the *Defining a Memory Map* chapter in the *TMS320C80 (MVP) C Source Debugger User's Guide*.

□ † *init.clr* is a general-purpose screen configuration file. If this file isn't present when you invoke the debugger, the debugger uses the default screen configuration.

□ † The default configuration is for color monitors; an additional file, *mono.clr*, can be used for monochrome monitors. When you first start to use the debugger, the default screen configuration should be sufficient for your needs. Later, you may want to define your own custom configuration.

For information about these files and about setting up your own screen configuration, refer to the *Customizing the Debugger Display* chapter in the *TMS320C80 (MVP) C Source Debugger User's Guide*.

† Included as part of the debugger package

1.2 Step 1: Installing the Emulator Board in Your PC

This section contains the hardware installation information for the emulator.

Preparing the emulator board for installation

Before you install the emulator board, you must be sure that the board's switches are set to correctly identify the I/O space that the board can use. The emulator uses 32 bytes of the PC I/O space; two switches on the board identify this space.

Figure 1–1 shows where these switches are on the emulator and identifies the switch numbers.

Switches are shipped in the default settings shown here and are listed in Table 1–1. If you use an I/O space that differs from the default, change the switch settings. Table 1–1 shows alternate settings.

In most cases, you can leave the switch settings in the default position. However, you must ensure that the 'C8x emulator I/O space does not conflict with other bus settings. For example, if you've installed a bus mouse in your system, you may not be able to use the default switch settings for the I/O space—the mouse might use this space. Refer to your PC technical reference manual and your other hardware-board manuals to see if there are any I/O space conflicts. If you find a conflict, use one of the alternate settings shown in Table 1–1.

Figure 1–1. Emulator Board I/O Switches

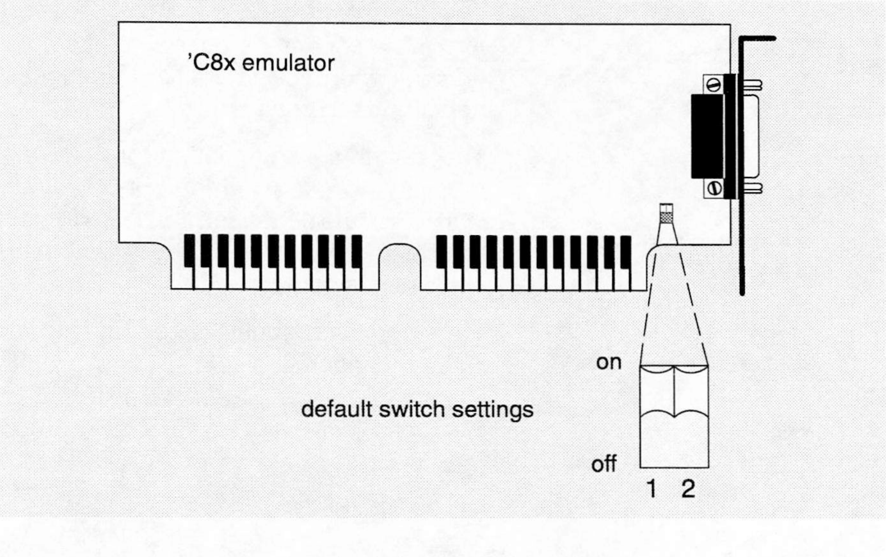


Table 1–1. Emulator Board Switch Settings

		Switch #	
	Address Range	**1**	**2**
default	0x0240–0x025F	on	on
	0x0280–0x029F	on	off
	0x0320–0x033F	off	on
	0x0340–0x035F	off	off

Some of the other installation steps require you to know which switch settings you used. If you reset the I/O switches, note the modified settings here for later reference.

Table 1–2. Your Switch Settings

	Switch #	
Address Range	**1**	**2**

WARNING

1) **To minimize the risk of electric shock and fire hazard, be sure that all major components that you interface with Texas Instruments devices are limited in energy and certified by one or more of the following agencies: UV, CSA, VDE, or TUV.**
2) **To minimize the risk of personal injury, *always* turn off the power to your PC and unplug the power cord before installing the 'C8x emulator board.**

Setting the emulator board into your PC

After you've prepared the emulator board for installation, follow these steps:

1) Turn off the power to the PC, and unplug the power cord.
2) Remove the cover of your PC.
3) Remove the mounting bracket from an unused 16-bit ISA/EISA bus slot.
4) Install the emulator board in the slot (see Figure 1–2).
5) Tighten down the mounting bracket.
6) Plug the emulator target cable into the emulator board (see Figure 1–3). The cable is a 25-pin DSUB connector, shaped to ensure proper connection.
7) Replace the PC cover.
8) Plug in the power cord, and turn on the power to the PC.

Figure 1–2. Emulator Board Installation

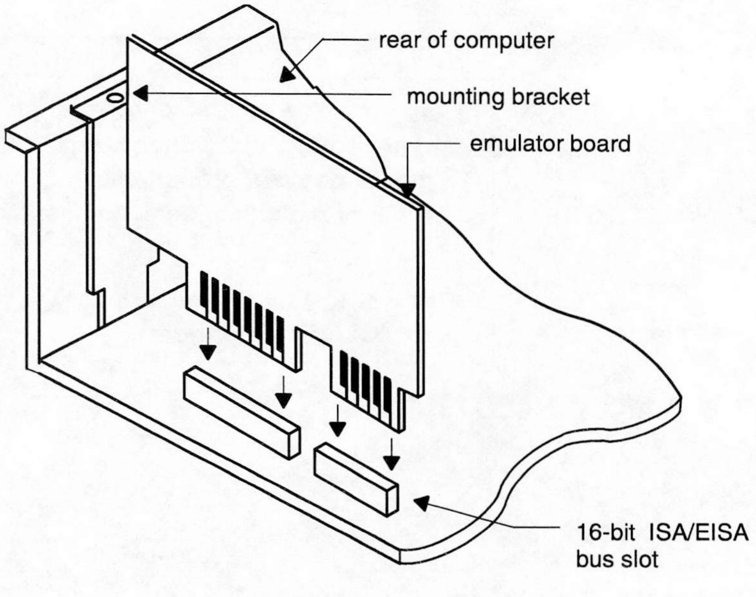
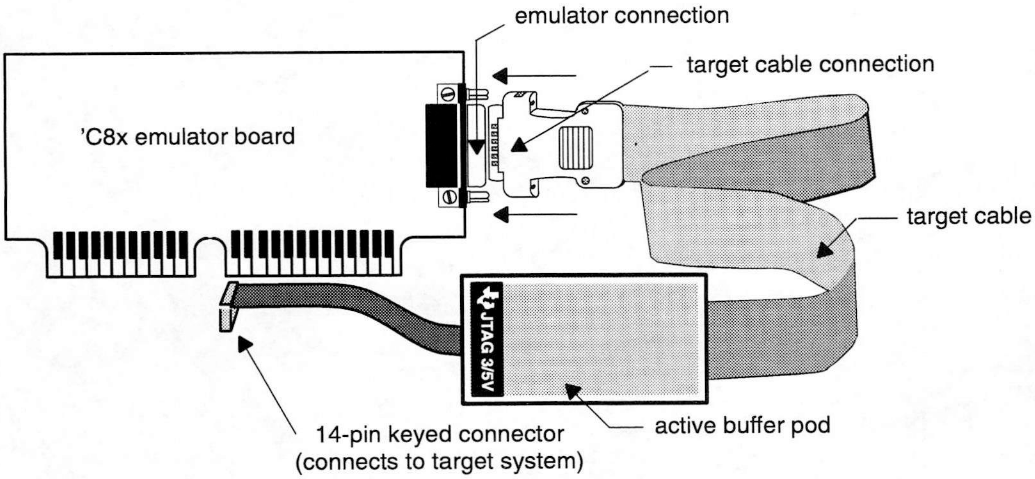


Figure 1–3. Emulator Target Cable and Board



CAUTION

Don't connect or disconnect the target cable to or from the 'C8x emulator board while the PC is powered up.

Be very careful with the target cable connectors. Connect them gently; forcing the connectors into position may damage them.

Remember, the connector is keyed. Be sure to connect the cable so that the key fits into its slot.

1.3 Step 2: Connecting the Emulator to Your Target System

Figure 1–4 shows a typical setup using the emulator, target cable, and your target system.

Figure 1–5 shows how you connect the emulator and target cable to your target system. In most cases, the target system will be a 'C8x board of your own design.

Figure 1–4. Typical Setup Using the 'C8x Emulator and Your Target System

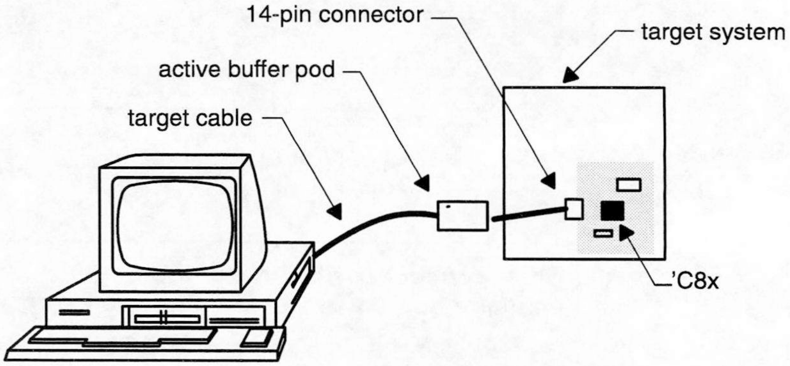
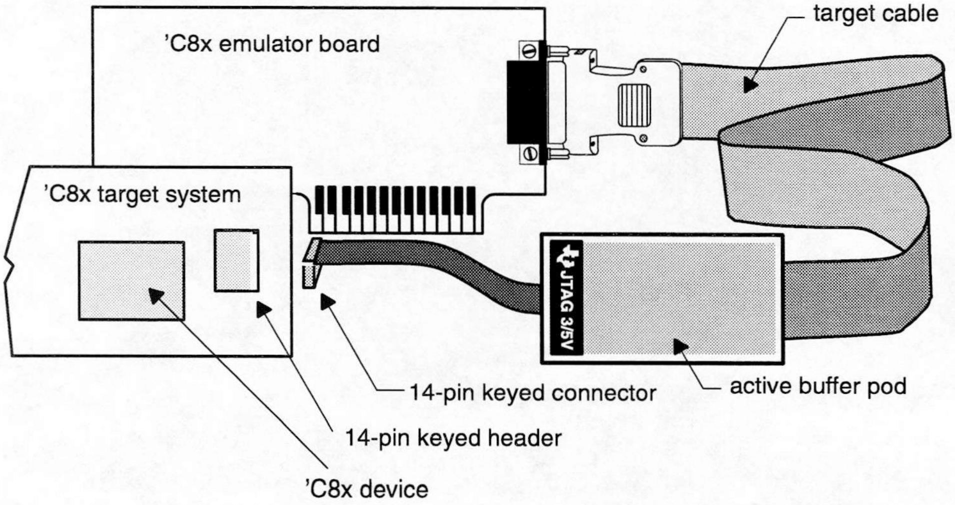


Figure 1–5. Connecting the 'C8x Emulator to Your Target System



1.4 Step 3: Installing the Debugger Software

This section explains the process of installing the debugger software on a hard-disk system.

1) On your hard disk, create a directory named *c8xhll.* This directory will contain the 'C8x C source debugger software. To create this directory, enter:

   ```
   MD C:\C8XHLL ↵
   ```

2) Insert the *TMS320C8x Emulator* CD-ROM into your CD-ROM drive.

3) Change to the CD-ROM drive (replace D with the name of your CD-ROM drive):

   ```
   D: ↵
   ```

4) Copy all of the directories and files on the CD-ROM to the hard disk:

   ```
   XCOPY /S /E /V *.* C:\C8XHLL ↵
   ```

   The XCOPY command copies the directories and files recursively and keeps the directory structure intact.

1.5 Step 4: Installing the Emulator Device Driver

You must install the emulator device driver (genport.sys) in the Windows NT registry to enable the debugger to communicate with the emulator. To do so, you can use the regdrv driver installation program. The regdrv program performs two tasks:

- It copies genport.sys to the Windows NT drivers directory. In most cases, this directory is C:\WINDOWS\SYSTEM32\DRIVERS or C:\WINNT35\SYSTEM32\DRIVERS. However, the directory name may be different on your PC.
- It registers the genport.sys parameters in the Windows NT registry.

To execute the regdrv program and install the emulator device driver, follow these steps:

1) Set the current directory to the directory where you installed the emulator software:

   ```
   CD C:\C8XHLL ⏎
   ```

2) Execute the regdrv program:

   ```
   REGDRV ⏎
   ```

   The regdrv program prompts you to enter the directory where you installed the emulator software. It defaults to the current directory:

   ```
   Default path: <C:\C8XHLL>:
   ```

3) To accept the default pathname, press ⏎.

   Next, regdrv prompts you to register the emulator driver parameters:

   ```
   Do you want to register the emulator driver parameters? <y/n>:
   ```

   - If you enter **y**, regdrv checks to see if the emulator driver is already installed. If it is, regdrv prompts you to replace/update the driver with the new emulator driver.
   - If you enter **n**, regdrv exits without updating the emulator driver parameters in the Windows NT registry.

4) Type **y** and press ⏎.

   When the registration is complete, the regdrv program displays the status of the registration.

5) Reboot your PC.

   If you encounter error messages during the installation process, refer to the *Driver Installation Error Messages* discussion on page 1-17.

1.6 Step 5: Describing Your Target System to the Debugger

In order for the debugger to understand how you have configured your target system, you must supply the target configuration information in a file for the debugger to read.

- ❑ If you're using an emulation scan path that contains only one 'C8x device and no other devices, you can use the *board.dat* file that comes with the 'C8x emulator kit. This file describes to the debugger the single 'C8x device in the scan path and gives the 'C8x device the name MVP1. Since the debugger automatically looks for a file called board.dat in the current directory and in the directories specified with the D_DIR environment variable, you don't need to create your own board configuration file. Go to the next section.

- ❑ If you plan to use a different target system, you must follow these steps:

  Step 1: Create the board configuration file.

  Step 2: Translate the board configuration file to binary so that the debugger can read it.

  Step 3: Specify the configuration file when invoking the debugger.

  These steps are described in the *Describing Your Target System to the Debugger* appendix in the *TMS320C80 (MVP) C Source Debugger User's Guide.*

1.7 Step 6: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- ❑ Modify the PATH statement to identify the c8xhll directory.
- ❑ Define environment variables so that the debugger can find the files it needs.
- ❑ Identify any nondefault I/O space used by the emulator.
- ❑ Reset the emulator board.

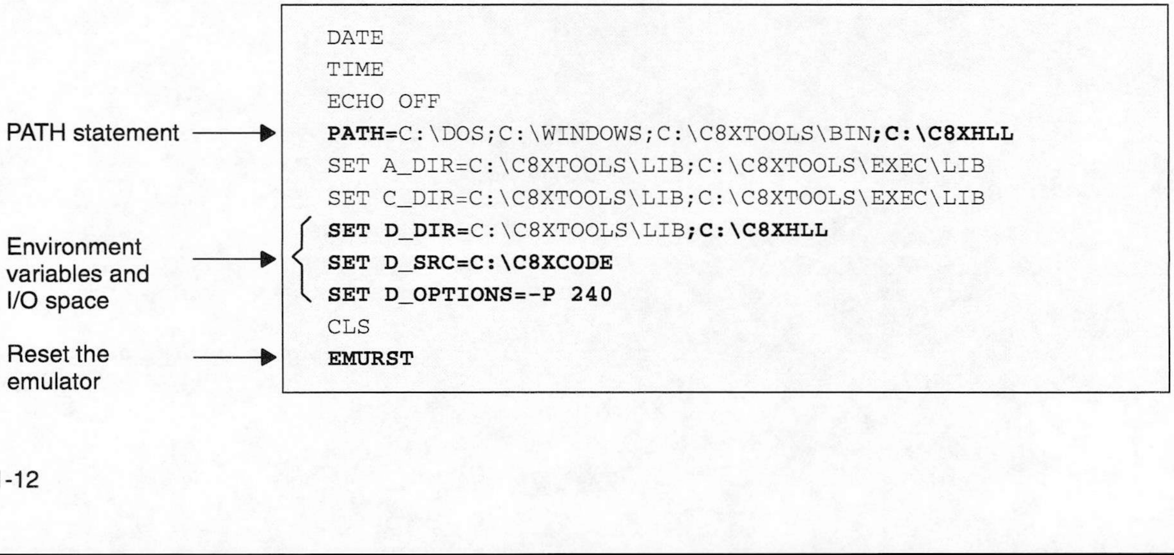
Not only must you do these things before you invoke the debugger for the first time, *you must do them any time you power up or reboot your PC.*

You can accomplish these tasks by entering individual commands, but it's simpler to put the commands in a batch file. You can edit your system's autoexec.bat file; in some cases, modifying the autoexec.bat file may interfere with other applications running on your PC. So, if you prefer, you can create a separate batch file that performs these tasks.

Figure 1–6 *(a)* shows an example of an autoexec.bat file that contains the suggested modifications (highlighted in bold type). Figure 1–6 *(b)* shows a sample batch file that you could create instead of editing the autoexec.bat file (for the purpose of discussion, assume that this sample file is named *initdb.bat*). The subsections following the figure explain these modifications.

Figure 1–6. Command Setup for the Debugger

(a) Sample autoexec.bat file to use with the debugger and emulator



```
DATE
TIME
ECHO OFF
PATH=C:\DOS;C:\WINDOWS;C:\C8XTOOLS\BIN;C:\C8XHLL
SET A_DIR=C:\C8XTOOLS\LIB;C:\C8XTOOLS\EXEC\LIB
SET C_DIR=C:\C8XTOOLS\LIB;C:\C8XTOOLS\EXEC\LIB
SET D_DIR=C:\C8XTOOLS\LIB;C:\C8XHLL
SET D_SRC=C:\C8XCODE
SET D_OPTIONS=-P 240
CLS
EMURST
```

Figure 1–6. Command Setup for the Debugger (Continued)

(b) Sample initdb.bat file to use with the debugger and emulator

PATH statement →

Environment variables and I/O space →

Reset the emulator →

```
PATH=C:\C8XHLL;%PATH%
SET D_DIR=C:\C8XHLL
SET D_SRC=C:\C8XCODE
SET D_OPTIONS=-P 240
EMURST
```

Invoking the new or modified batch file

- ☐ If you modify the autoexec.bat file, be sure to invoke it before invoking the debugger for the first time. To invoke this file, enter:

  AUTOEXEC ↵

- ☐ If you create an initdb.bat file, you must invoke it before invoking the debugger for the first time. You'll need to invoke initdb.bat any time that you power up or reboot your PC. To invoke this file, enter:

  INITDB ↵

Modifying the PATH statement

To invoke the debugger without specifying the name of the directory that contains the debugger executable file, define a path to the debugger directory. The general format is:

```
PATH=C:\C8XHLL
```

- ☐ If you are modifying an autoexec that already contains a PATH statement, simply include **;C:\c8xhll** at the end of the statement, as shown in Figure 1–6 *(a)*.

- ☐ If you are creating an initdb.bat file, use a different format for the PATH statement, as shown in Figure 1–6 *(b)*:

  ```
  PATH=C:\C8XHLL;%PATH%
  ```

  The addition of **;%path%** ensures that this PATH statement won't undo PATH statements in any other batch files (including the autoexec.bat file).

Setting up the environment variables

An environment variable is a special system symbol that the debugger uses for finding or obtaining certain types of information. The debugger uses three environment variables named D_DIR, D_SRC, and D_OPTIONS. The next three steps tell you how to set up these environment variables. The format for doing this is the same for both the autoexec.bat and initdb.bat files.

- ❑ Set up the D_DIR environment variable to identify the c8xhll directory:

  ```
  SET D_DIR=C:\C8XHLL
  ```

  (Be careful not to precede the equal sign with a space.)

  This directory contains auxiliary files (emurst, init.cmd, etc.) that the debugger needs.

- ❑ Set up the D_SRC environment variable to identify any directories that contain program source files that you'll want to look at while you're debugging code. The general format for doing this is:

  SET D_SRC=*pathname$_1$;pathname$_2$...*

  (Be careful not to precede the equal sign with a space.)

  For example, if your 'C8x programs were in a directory named *c8xcode* on drive C, the D_SRC setup would be:

  ```
  SET D_SRC=C:\C8XCODE
  ```

- ❑ You can use several options when you invoke the debugger. If you use the same options over and over, it's convenient to specify them with D_OPTIONS. The general format for doing this is:

  SET D_OPTIONS= [*object filename*] [*debugger options*]

  (Be careful not to precede the equal sign with a space.)

  This tells the debugger to load the specified object file and use the specified options each time you invoke the debugger. Table 1–3 lists the options that you can identify with D_OPTIONS

  Note that you can override D_OPTIONS by invoking the debugger with the –x option.

  For more information about options, see the *Debugger Options* section in the *Overview of a Code Development and Debugging System* chapter in the *TMS320C80 (MVP) C Source Debugger User's Guide.*

Table 1–3. Options for Use With D_OPTIONS

Option	Description
–b[b]	Select the screen size
–d *machinename*	Display debugger on different machine (X Windows only)
–f *filename*	Identify a new board configuration file
–i *pathname*	Identify additional directories
–n *processorname*	Identify the name of the processor
–o	Enable C I/O
–p *driver number*	Identify the device driver
–s	Load the symbol table only
–t *filename*	Identify a new initialization file
–v	Load without the symbol table

Identifying the correct I/O switches

Refer to your entries in Table 1–2 (page 1-5). If you didn't modify the I/O switches, skip this step.

If you modified the I/O switch settings, you must use the debugger's –p option to identify the I/O space that the emulator is using. You can do this each time you invoke the debugger, or you can specify this information by using the D_OPTIONS environment variable. Table 1–4 lists the nondefault I/O switch setting and the appropriate line that you can add to the autoexec.bat or initdb.bat file.

Table 1–4. Identifying Nondefault I/O Address Space

	switch #		
Address Range	**1**	**2**	**Add this line to the batch file**
0x0280–0x029F	on	off	`SET D_OPTIONS=-p 280`
0x0320–0x033F	off	on	`SET D_OPTIONS=-p 320`
0x0340–0x035F	off	off	`SET D_OPTIONS=-p 340`

Resetting the emulator

To reset the 'C8x emulator, add this line to the autoexec.bat or initdb.bat file:

```
emurst
```

> **Note:**
>
> The emurst program resets the 'C8x emulator only; it doesn't apply reset to the 'C8x device on your target board. If you cycle the power to your target board, you **must** execute emurst before you invoke the debugger.

1.8 Step 7: Verifying the Emulator Driver Installation

To ensure that you have correctly installed the emulator driver, follow these steps:

1) Turn off the power to the 'C8x target board.
2) Connect the JTAG cable from your PC to the 'C8x target board.
3) Turn on the power to the 'C8x target board.
4) If a hard RESET is not generated automatically after power-up, apply a hard RESET to the 'C8x device.
5) From the command prompt in a MS-DOS™ window, enter:

   ```
   emurst ↵
   ```

You should see a message similar to the following:

```
XDS510 IS RESET, HARDWARE VERSION 3
```

□ If this message appears, you have correctly installed the emulator driver. Turn to Section 1.9, *Step 8: Verifying the Emulator and Debugger Installation.*

□ If you see this message:

```
CANNOT DETECT TARGET POWER
```

follow the troubleshooting tips in the following sections.

Troubleshooting

□ The target power is not on or the cables are not connected.

Be sure that power is applied to the target board and that the JTAG cable is connected firmly.

□ The I/O switch settings on the 'C8x emulator board don't match the value specified with the –p option.

If you are using nondefault I/O switch settings, you must specify the corresponding –p option in either the D_OPTIONS environment variable or on the command line when you invoke the debugger.

- ☐ The emulator driver is not installed properly.

  See the error messages listed in the *Driver Installation Error Messages* subsection (page 1-17) to determine a course of action.

Driver installation error messages

Invalid Source path

Description The directory path entered does not exist.

Action Be sure that the emulator software directory has been correctly entered. Discontinue the driver installation program by entering **n** at the next prompt. Reinvoke the regdrv program and reenter the emulator software directory.

Can't find the emulator driver

Description The directory path that you entered does not contain the emulator driver genport.sys.

Action Check directory path and make sure it corresponds to the emulator software directory.

Can't find Windows NT SYSTEM32\DRIVERS directory

Description The Windows NT drivers directory should be c:\pathname\system32\drivers. Typically, the *pathname* is c:\windows or c:\winnt35.

Action Reinvoke the regdrv program and enter the correct pathname to the system32\drivers directory. If you continue to see this message, you may need to install the emulator driver manually. Refer to Appendix A, *Installing the Emulator Driver Manually,* for more information.

Can't overwrite emulator driver – access denied

Description The permissions for the emulator driver are not correct.

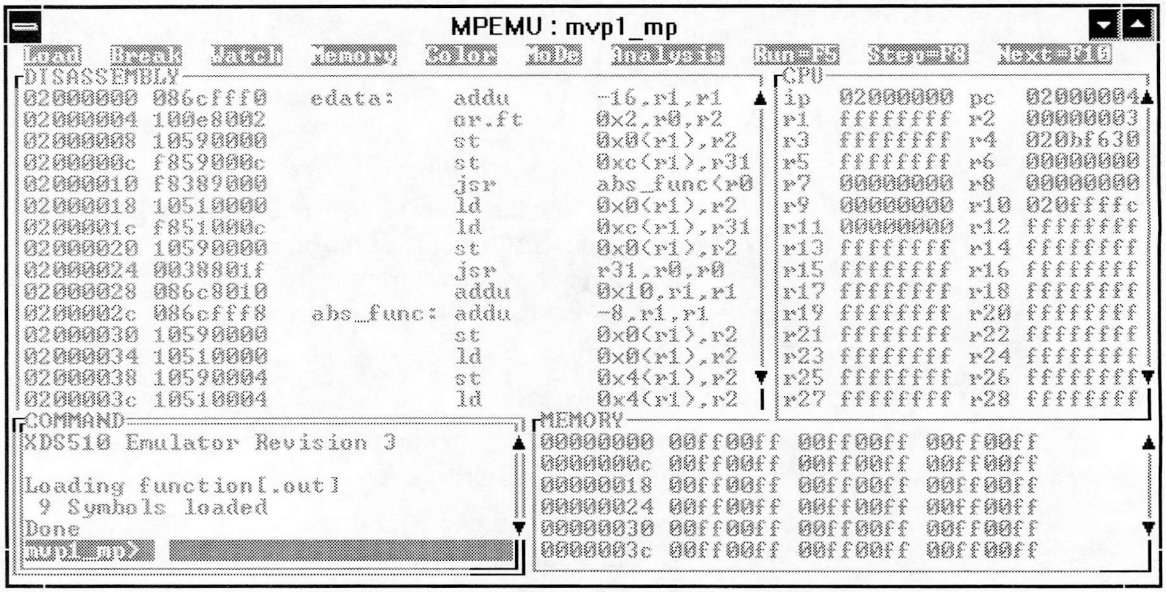
Action Check file permissions on emulator driver that is already installed. If driver is loaded in memory, from the command line, type *NET STOP GENPORT* to stop the driver. Reinstall the driver by following the directions in Section 1.5.

1.9 Step 8: Verifying the Emulator and Debugger Installation

To ensure that you have correctly installed the emulator and debugger software, enter this command at the system prompt:

```
mpemu c:\c8xhll\function -n MVP1_MP
```

You should see a display similar to this one:



```
DISASSEMBLY
02000000 086cfff0   edata:     addu    -16,r1,r1
02000004 100e8002              or.ft   0x2,r0,r2
02000008 10590000              st      0x0(r1),r2
0200000c f859000c              st      0xc(r1),r31
02000010 f8389000              jsr     abs_func(r0
02000018 10510000              ld      0x0(r1),r2
0200001c f851000c              ld      0xc(r1),r31
02000020 10590000              st      0x0(r1),r2
02000024 0038801f              jsr     r31,r0,r0
02000028 086c8010              addu    0x10,r1,r1
0200002c 086cfff8   abs_func:  addu    -8,r1,r1
02000030 10590000              st      0x0(r1),r2
02000034 10510000              ld      0x0(r1),r2
02000038 10590004              st      0x4(r1),r2
0200003c 10510004              ld      0x4(r1),r2
```

```
CPU
ip  02000000 pc  02000004
r1  ffffffff r2  00000003
r3  ffffffff r4  020bf630
r5  ffffffff r6  00000000
r7  00000000 r8  00000000
r9  00000000 r10 020ffffc
r11 00000000 r12 ffffffff
r13 ffffffff r14 ffffffff
r15 ffffffff r16 ffffffff
r17 ffffffff r18 ffffffff
r19 ffffffff r20 ffffffff
r21 ffffffff r22 ffffffff
r23 ffffffff r24 ffffffff
r25 ffffffff r26 ffffffff
r27 ffffffff r28 ffffffff
```

```
COMMAND
XDS510 Emulator Revision 3

Loading function[.out]
 9 Symbols loaded
Done
mvp1_mp>
```

```
MEMORY
00000000 00ff00ff 00ff00ff 00ff00ff
0000000c 00ff00ff 00ff00ff 00ff00ff
00000018 00ff00ff 00ff00ff 00ff00ff
00000024 00ff00ff 00ff00ff 00ff00ff
00000030 00ff00ff 00ff00ff 00ff00ff
0000003c 00ff00ff 00ff00ff 00ff00ff
```

□ If you see a display similar to this one, you have correctly installed your emulator and debugger.

□ If you see a display and the lines of code say *Invalid address* or the fields in the MEMORY window are shown in red, the debugger may not be able to find the init.cmd file. Check for the file in the directories specified by the D_DIR environment variable or ensure that the file is in the current directory. Re-enter the command above.

□ If you don't see a display, then your debugger or emulator board may not be installed properly. Go back through the installation instructions and be sure that you have followed each step correctly; then re-enter the command above.

Installation error messages

While invoking the debugger, you may see the following message:

```
CANNOT INITIALIZE THE TARGET !!
- Check I/O configuration
- Check cabling and target power
Init error -1
```

One of several of the following conditions may be the cause; check:

- □ Is the target power on?
- □ Is the emulator board installed firmly?
- □ Is the cable connecting your emulator and target system loose?
- □ Is your target board getting the correct voltage?
- □ Is your port address set correctly:
- □ Does the emurst command appear at the end of either your autoexec.bat or initdb.bat file? This command must be executed *after* you powered up the target board.
  - Check to be sure the –p option used with the D_OPTIONS environment variable matches the I/O address defined by your switch settings (refer to *Your Switch Settings*, Table 1–2, and *Identifying Nondefault I/O Address Space*, Table 1–4).
  - Check to see if you have a conflict in address space with another bus setting. If you have a conflict, change the switches on your board to one of the alternate settings in Table 1–1. Modify the –p option of the D_OPTIONS environment variable to reflect the change in your switch settings.
- □ Was the –f option specified on the command line or in the D_OPTIONS environment variable (where the default file specified by the –f option is board.dat) specifying a file that the debugger can't find?
  - If you didn't provide *any* path information with the filename, the debugger couldn't find the file in the current directory or in any of the directories listed in the D_DIR environment variable.
  - If you didn't provide the *correct* path information, re-execute the debugger, specifying the correct pathname and filename for the board configuration file.

- □ Was the **–n MVP1_MP** option specified when you invoked the debugger? You *must* use the –n option so that the debugger knows which 'C8x device on your target system you want to debug.
- □ Was the processor name specified with the **–n option** in your board configuration file?

After you have checked all of the above, repeat the verification instructions in Section 1.9.

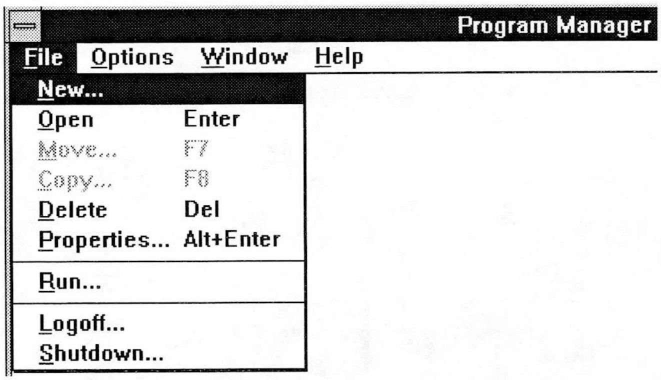
1.10 Setting Up Icons for the Windows NT Program Manager

There are two ways to invoke the debugger. You are already familiar with the first method; you can invoke the debugger from the MS-DOS command line:

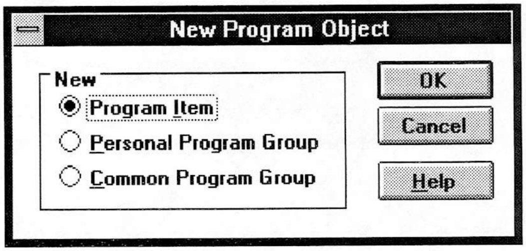
```
mpemu -nmvp1_mp ⏎
```

An easier way to invoke the debugger is to establish a program group window that contains an icon for each of the 'C8x debuggers. This allows you to invoke a debugger by simply double-clicking on its icon.

Step 1: From the Windows NT Program Manager, select File → New.



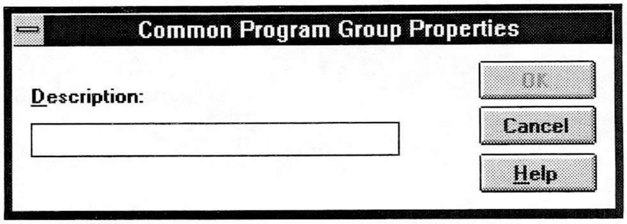
This displays the New Program Object dialog box:



New
● Program Item
○ Personal Program Group
○ Common Program Group

Step 2: Select the type of program group you want—Personal Program Group or Common Program Group—then click on the OK button.

This displays a dialog box for you to enter the group name:



Description:

Step 3: Enter *TMS320C8x Debuggers* as the Description, then click on the OK button.

This displays an empty program group named TMS320C8x Debuggers.

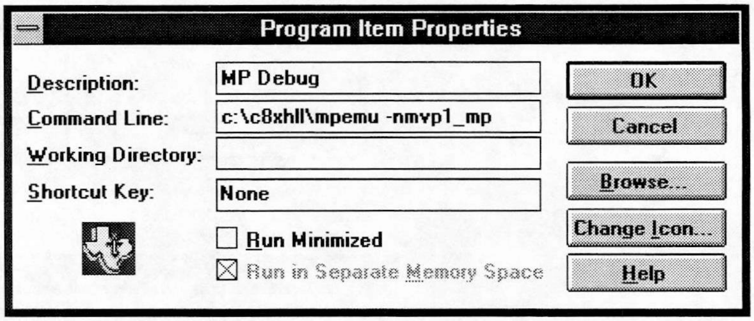
Now you can assign program items (debugger icons) to the TMS320C8x Debuggers program group. Be sure that the TMS320C8x Debuggers program group is active. If it is not highlighted (active), move your mouse into its window and click.

Step 4: While the TMS320C8x Debuggers program group is active, from the Windows NT Program Manager, select File → New.

This displays the New Program Object dialog box.

Step 5: Select the Program Item option button, then click on the OK button.

This displays the Program Item Properties dialog box. Enter the following information for the MP debugger:



Program Item Properties

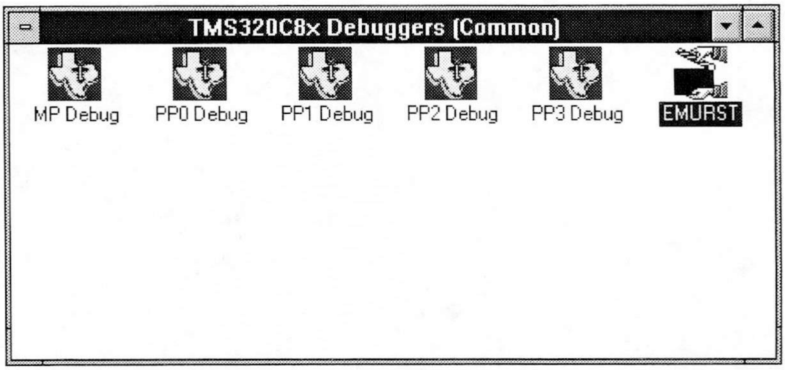
Description: MP Debug
Command Line: c:\c8xhll\mpemu -nmvp1_mp
Working Directory:
Shortcut Key: None
☐ Run Minimized
☒ Run in Separate Memory Space

After entering this information, click on the OK button. This displays a Texas Instruments icon in your TMS320C8x Debuggers program group.

Step 6: Repeat steps 4 and 5 for each of the PP debuggers. You can also put emurst into the TMS320C8x program group. The program item descriptions and command lines are:

Description	Command Line
EMURST	c:\c8xhll\emurst –p 240
PP0 Debug	c:\c8xhll\ppemu –n mvp1_pp0
PP1 Debug	c:\c8xhll\ppemu –n mvp1_pp1
PP2 Debug	c:\c8xhll\ppemu –n mvp1_pp2
PP3 Debug	c:\c8xhll\ppemu –n mvp1_pp3

After entering all the debugger program items, the TMS320C8x Debuggers program group should look like the following:



Step 7: Invoke a debugger by double-clicking on its icon.

Chapter 2

Release Notes

This chapter contains documentation of features that are new or have been changed since the last release. This is not an exhaustive list of all code changes since the last release, but it is a list of all of the changes that may require modifications to your debugger batch or initialization files, or makefiles, so please take time to read this chapter completely.

Topic **Page**

2.1 Release Enhancements

C I/O support (–o option)

If to plan to debug a program that uses the standard C I/O facilities, you must invoke the MP or PP debugger with the –o option. When you use the –o option, the debugger sets up a special environment with the host machine for performing I/O functions. This environment will not be set up if you don't use the –o option.

The –o option was added to prevent more than one debugger from attempting to set up the special C I/O environment when multiple 'C8x debuggers are running. Since only one debugger (or processor) can be using the C I/O facilities at a time, the –o option allows you to explicitly identify which debugger will set up the environment.

This change may require you to modify the PDM initialization files or script files that you use to invoke the debuggers.

Global breakpoints

You can use global breakpoints, through the use of the EMU0/1 pins (when connected as shown in the *JTAG/MPSD Emulation Technical Reference*), to halt multiple processors across multiple 'C8x devices. The EMU0 pin is also used as a carry out for the RUNB counter. See the *Using the Analysis Interface* chapter in the *TMS320C80 (MVP) C Source Debugger User's Guide* for details on global breakpoints.

A Breakpoints Disable selection has been added to the **Analysis break events** dialog box, allowing you to disable the bus breakpoints without disabling the **EMU0/1 driven low** breakpoint.

Little-endian support

Both MP and PP debuggers now support little-endian 'C8x operation. Before invoking a debugger (MP or PP), you must reset to the 'C8x device (through the $\overline{\text{RESET}}$ pin) to set the 'C8x device's endian order. To change the endian order of the 'C8x device, you must quit all debuggers that are active, reset the 'C8x device, then restart the debuggers. See the *TMS320C80 Data Sheet* for information about the $\overline{\text{RESET}}$ pin.

When the 'C8x device is configured for big-endian operation, the address of values in the MEMORY window are referenced from the MSB. If the 'C8x device is configured for little-endian operation, the address of values in the MEMORY window are referenced from the LSB.

Memory accesses less than a word

In previous revisions of the emulator, all memory accesses were performed on word or doubleword address boundaries. If an access was performed that required less than a word (byte or halfword):

- ❑ For a read operation, the emulator performed a word read and then extracted the required data.
- ❑ For a write operation, the emulator performed a word read, modified the data, and then performed a word write.

This version of the emulator accesses only the addresses requested. If a byte or halfword is requested, the emulator performs the appropriate memory operation to access only the requested data. For example:

```
?*(char *)0x02000000 = 0x12
```

For this example, previous revisions performed a word read from memory, modified the word with the byte data, and then performed a word write. The current revision simply performs a byte write of the data.

Memory access alignment

The emulator now automatically aligns accesses to the 'C8x device's natural address boundaries. If you request a short data type on an odd-byte address boundary, the emulator performs two byte accesses. If you request an int (integer) data type on an odd-byte address, the emulator performs a byte access, a halfword access, followed by a byte access.

Viewing the data cache (CACHEVIEW command)

The CACHEVIEW command has been added for this release. This command toggles the view of all memory accesses generated by the debugger between the data cache view and the external memory view.

- ❑ When the data cache view is enabled (default), memory writes are performed to both the cache and the external memory location (cache write through).
- ❑ When the external memory view is enabled, memory writes are performed only with the external memory location.

Previous revisions of the MP debugger accessed memory from the data caches first, then from external memory. This feature restricted access to external memory when data was found within the MP's data cache.

Disabling caching

When the emulator is performing memory operations, it caches memory accesses. If you request a byte, the emulator may read up to eight full words to service its internal cache, and then extract the byte from that data. To disable caching for memory locations that you want to access individually (such as a memory mapped I/O port), use the MA command to map the address as an I/O port. See the *Defining a Memory Map* chapter in the *TMS32080 (MVP) C Source Debugger User's Guide* for more information on I/O port mapping.

Multiprocessor support

This revision of the emulator supports any combination of 1.x, 2.0, and 3.0 devices on a scan chain.

2.2 Special Notes About the RUNB Command

You can use the RUNB command to benchmark a specific section of code. RUNB counts clocks until execution is halted by a software or analysis breakpoint. You can use the CLK pseudoregister to display the count value in the COMMAND window (? command) or in a WATCH window. You can also use it in an expression. The value of CLK is only accurate when you execute code from one breakpoint to another.

RUNB is enabled only for preproduction silicon release 2 devices and higher.

For preproduction silicon release 2, the PP's instruction cache is flushed before every execute command, which causes a cache miss to occur as the command starts to execute. The PP's CLK value includes the number of clocks the extra cache miss took to service as well as the number of clocks the instruction took to execute. The cache flush is a fix for the preproduction silicon release 2 PP instruction-cache scan problem; see the emulation errata file on the *TMS320C8x Emulator* CD-ROM for more information.

Appendix A

Installing the Emulator Device Driver Manually

You must install the emulator device driver (genport.sys) in the Windows NT registry to enable the debugger to communicate with the emulator. If you can't get regdrv installation program (described in Section 1.5 on page 1-10) to install and register the emulator device driver, you can install the driver manually by performing the following steps:

1) Copy the device driver file and invoke the registry editor.
2) Set up the Genport directory.
3) Set up the values for the Genport directory.
4) Set up the Parameters directory and values.

This appendix describes these steps.

Topic **Page**

A.1 Copying the Device Driver File and Invoking the Registry Editor

The *TMS320C8x Emulator* CD-ROM includes a genport.sys file that contains information about the emulator device driver. You must copy this file to the Windows NT drivers directory. Once you do so, you can use the registry editor to register the genport.sys parameters, allowing the emulator and debugger to communicate.

1) Copy genport.sys to the Windows NT drivers directory. For example:

   ```
   copy c:\c8xhll\genport.sys c:\winnt35\system32\drivers ⏎
   ```

   or

   ```
   copy c:\c8xhll\genport.sys c:\windows\system32\drivers ⏎
   ```

2) Invoke the registry editor:

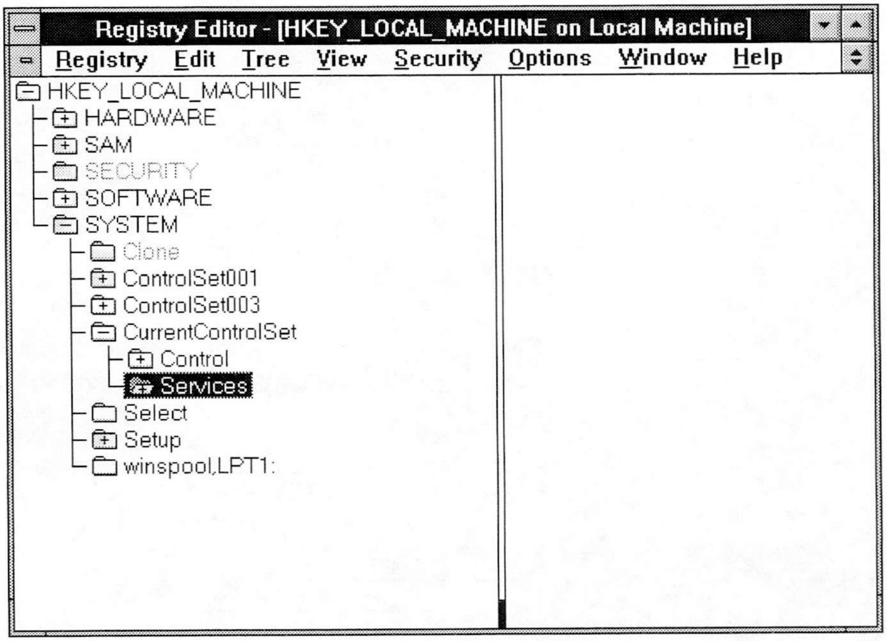
   ```
   regedt32 ⏎
   ```

   The registry editor brings up several windows.

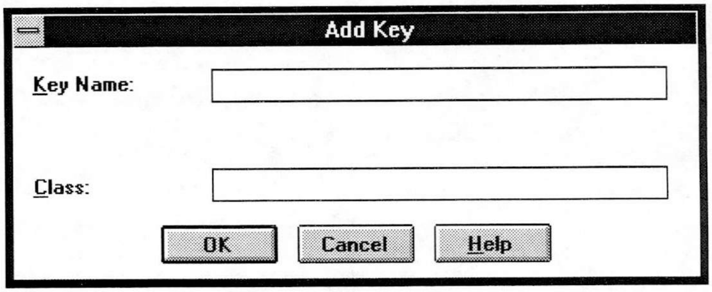
The remaining sections in this appendix tell you how to register the genport.sys parameters.

A.2 Setting Up the Genport Directory

1) After you invoke the registry editor, locate the window called *HKEY_LOCAL_MACHINE on Local Machine* and select the SYSTEM\CurrentControlSet\Services directory icon.



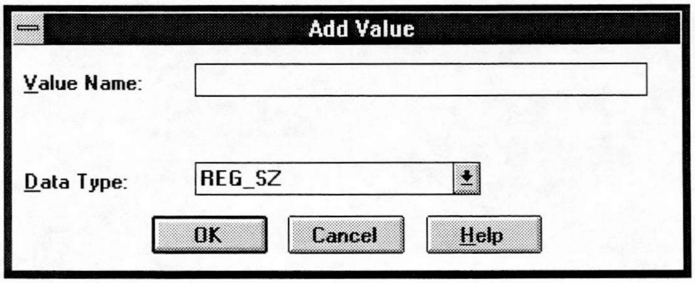
2) While the Services directory icon is highlighted, select Edit → Add Key from the menu bar. This displays the Add Key dialog box:



3) Enter *Genport* as the Key Name, then click on the OK button. This causes the Genport directory icon to appear under the Services directory (you may need to scroll the window to see the Genport directory).

A.3 Setting Up the Values for the Genport Directory

1) Click on the Genport directory icon to make it active (highlighted), then select Edit → Add Value from the menu bar. This displays the Add Value dialog box:

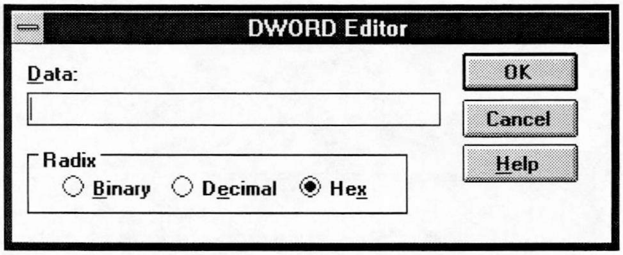


Add Value

Value Name:

Data Type: REG_SZ

2) Enter *ErrorControl* as the Value Name, select *REG_DWORD* as the Data Type, then click on the OK button. This displays the DWORD editor dialog box:



DWORD Editor

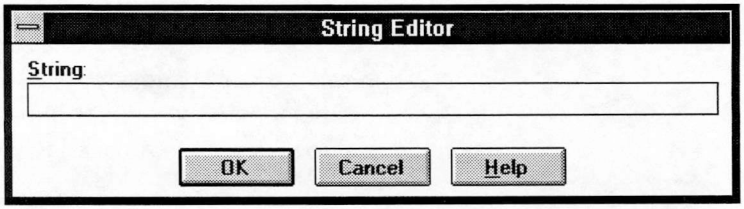
Data:

Radix ○ Binary ○ Decimal ● Hex

3) Enter *1* as the Data, select *Hex* as the Radix, then click on the OK button.

4) Click on the Genport directory icon to make it active (highlighted), then select Edit → Add Value from the menu bar. This displays the Add Value dialog box.

5) Enter *Group* as the Value Name, select *REG_SZ* as the Data Type, then click on the OK button. This displays the String Editor dialog box:



String Editor

String:

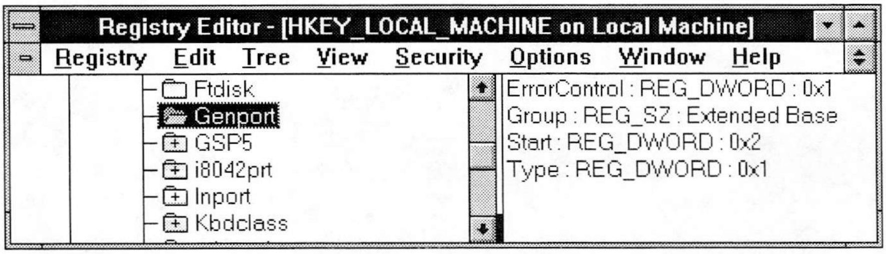
6) Enter *Extended Base* as the String, then click on the OK button.

7) Repeat steps 1–3 with the following information:

Value Name:	Start
Data Type:	REG_DWORD
Data:	2
Radix:	Hex

Value Name:	Type
Data Type:	REG_DWORD
Data:	1
Radix:	Hex

The following values should appear in the registry editor when the Genport directory icon is highlighted:



ErrorControl : REG_DWORD : 0x1
Group : REG_SZ : Extended Base
Start : REG_DWORD : 0x2
Type : REG_DWORD : 0x1

A.4 Setting Up the Parameters Directory and Values

1) Click on the Genport directory icon to make it active, then select Edit → Add Key from the menu bar. This displays the Add Key dialog box.

2) Enter *Parameters* as the Key Name, then click on the OK button. This causes the Parameters directory icon to appear under the Genport directory.

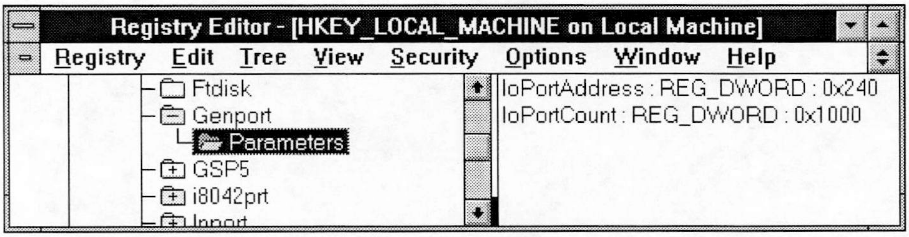
3) Click on the Parameters directory to make it active, select Edit → Add Value from the menu bar, then enter the following information:

Value Name:	IoPortAddress
Data Type:	REG_DWORD
Data:	240
Radix:	Hex

4) Repeat step 3 with the following information:

Value Name:	IoPortCount
Data Type:	REG_DWORD
Data:	1000
Radix:	Hex

The following values should appear in the registry editor when the Parameters directory icon is highlighted:



IoPortAddress : REG_DWORD : 0x240
IoPortCount : REG_DWORD : 0x1000

5) Exit the registry editor, then reboot your PC. This completes the manual driver installation process.

Index

NOTES

TI Worldwide Sales and Representative Offices

AUSTRALIA / NEW ZEALAND: Texas Instruments Australia Ltd.: Sydney [61] 2-910-3100, Fax 2-805-1186; **Melbourne** 3-696-1211, Fax 3-696-4446.
BELGIUM: Texas Instruments Belgium S.A./N.V.: Brussels [32] (02) 726-75-80, Fax (02) 726 72 76.
BRAZIL: Texas Instrumentos Electronicos do Brasil Ltda.: Sao Paulo [55] 11-535-5133.
CANADA: Texas Instruments Canada Ltd.: Montreal (514) 335-8392; **Ottawa** (613) 726-3201; **Toronto** (416) 884-9181.
DENMARK: Texas Instruments A/S: Ballerup [45] (44) 68 74 00.
FINLAND: Texas Instruments/OY: Espoo [358] (0) 43 54 20 33, Fax (0) 46 73 23.
FRANCE: Texas Instruments France: Velizy-Villacoublay Cedex [33] (1) 30 70 10 01, Fax (1) 30 70 10 54.
GERMANY: Texas Instruments Deutschland GmbH.: Freising [49] (08161) 80-0, Fax (08161) 80 45 16; **Hannover** (0511) 90 49 60, Fax (0511) 64 90 331; **Ostfildern** (0711) 34 03 0, Fax (0711) 34 032 57.
HONG KONG: Texas Instruments Hong Kong Ltd.: Kowloon [852] 2956-7288, Fax 2965-2200.
HUNGARY: Texas Instruments Representation: Budapest [36] (1) 269 8310, Fax (1) 267 1357.
IRELAND: Texas Instruments Ireland Ltd.: Dublin [353] (01) 475 52 33, Fax (01) 478 14 63.
ITALY: Texas Instruments Italia S.p.A.: Agrate Brianza [39] (039) 68 42.1, Fax (039) 68 42.912; **Rome** (06) 657 26 51.
JAPAN: Texas Instruments Japan Ltd.: Tokyo [81] 03-769-8700, Fax 03-3457-6777; **Osaka** 06-204-1881, Fax 06-204-1895; **Nagoya** 052-583-8691, Fax 052-583-8696; **Ishikawa** 0762-23-5471, Fax 0762-23-1583; **Nagano** 0263-33-1060, Fax 0263-35-1025; **Kanagawa** 045-338-1220, Fax 045-338-1255; **Kyoto** 075-341-7713, Fax 075-341-7724; **Saitama** 0485-22-2440,
Fax 0425-23-5787; **Oita** 0977-73-1557, Fax 0977-73-1583.
KOREA: Texas Instruments Korea Ltd.: Seoul [82] 2-551-2800, Fax 2-551-2828.
MALAYSIA: Texas Instruments Malaysia Sdn Bhd: Kuala Lumpur [60] 3-208-6001, Fax 3-230-6605.
MEXICO: Texas Instruments de Mexico S.A. de C.V.: Colonia del Valle [52] 5-639-9740.
NORWAY: Texas Instruments Norge A/S: Oslo [47] (02) 264 75 70.
MAINLAND CHINA: Texas Instruments China Inc.: Beijing [86] 1-500-2255, Ext. 3750, Fax 1-500-2705.
PHILIPPINES: Texas Instruments Asia Ltd.: Metro Manila [63] 2-817-6031, Fax 2-817-6096.
PORTUGAL: Texas Instruments Equipamento Electronico (Portugal) LDA.: Maia [351] (2) 948 10 03, Fax (2) 948 19 29.
SINGAPORE (& INDIA, INDONESIA, THAILAND): Texas Instruments Singapore (PTE) Ltd.: Singapore [65] 390-7100, Fax 390-7062.
SPAIN: Texas Instruments España S.A.: Madrid [34] (1) 372 80 51, Fax (1) 372 82 66; **Barcelona** (3) 31 791 80.
SWEDEN: Texas Instruments International Trade Corporation (Sverigefilialen): Kista [46] (08) 752 58 00, Fax (08) 751 97 15.
SWITZERLAND: Texas Instruments Switzerland AG: Dietikon [41] 886-2-3771450.
TAIWAN: Texas Instruments Taiwan Limited: Taipei [886] (2) 378-6800, Fax 2-377-2718.
UNITED KINGDOM: Texas Instruments Ltd.: Northampton [44] (0234) 270 111, Fax (0234) 223 459.

UNITED STATES: Texas Instruments Incorporated: ALABAMA: Huntsville (205) 430-0114; **ARIZONA: Phoenix** (602) 224-7800; **CALIFORNIA: Irvine** (714) 660-1200; **San Diego** (619) 278-9600; **San Jose** (408) 894-9000; **Woodland Hills** (818) 704-8100; **COLORADO: Denver** (303) 488-9300; **CONNECTICUT: Wallingford** (203) 265-3807; **FLORIDA: Orlando** (407) 667-5300; **Fort Lauderdale** (305) 425-7820; **Tampa** (813) 882-0017; **GEORGIA: Atlanta** (404) 662-7967; **ILLINOIS: Arlington Heights** (708) 640-2925; **INDIANA: Indianapolis** (317) 573-6400; **KANSAS: Kansas City** (913) 451-4511; **MARYLAND: Columbia** (410) 312-7900; **MASSACHUSETTS: Boston** (617) 895-9100; **MICHIGAN: Detroit** (313) 553-1500; **MINNESOTA: Minneapolis** (612) 828-9300; **NEW JERSEY: Edison** (908) 906-0033; **NEW MEXICO: Albuquerque** (505) 345-2555; **NEW YORK: Poughkeepsie** (914) 897-2900; **Long Island** (516) 454-6601; **Rochester** (716) 385-6770; **NORTH CAROLINA: Charlotte** (704) 522-5487; **Raleigh** (919) 876-2725; **OHIO: Cleveland** (216) 765-7258; **Dayton** (513) 427-6200; **OREGON: Portland** (503) 643-6758; **PENNSYLVANIA: Philadelphia** (215) 825-9500; **PUERTO RICO: Hato Rey** (809) 753-8700; **TEXAS: Austin** (512) 250-6769; **Dallas** (214) 917-1264; **Houston** (713) 778-6592; **WISCONSIN: Milwaukee** (414) 798-1001.

North American Authorized Distributors

COMMERCIAL	
Almac / Arrow	800-426-1410 / 800-452-9185 Oregon only
Anthem Electronics	800-826-8436
Arrow / Schweber	800-777-2776
Future Electronics (Canada)	800-388-8731
Hamilton Hallmark	800-332-8638
Marshall Industries	800-522-0084 or www.marshall.com
Wyle	800-414-4144
OBSOLETE PRODUCTS	
Rochester Electronics	508-462-9332
MILITARY	
Alliance Electronics Inc	800-608-9494
Future Electronics (Canada)	800-388-8731
Hamilton Hallmark	800-332-8638
Zeus, An Arrow Company	800-524-4735
CATALOG	
Allied Electronics	800-433-5700
Arrow Advantage	800-777-2776
Newark Electonics	800-367-3573

For Distributors outside North America, contact your local Sales Office.

A040395

Important Notice: Texas Instruments (TI) reserves the right to make changes to or to discontinue any product or service identified in this publication without notice. TI advises its customers to obtain the latest version of the relevant information to verify, before placing orders, that the information being relied upon is current.

Please be advised that TI warrants its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. TI assumes no liability for applications assistance, software performance, or third-party product information, or for infringement of patents or services described in this publication. TI assumes no responsibility for customers' applications or product designs.

Printed in the USA

TEXAS
INSTRUMENTS

TEXAS INSTRUMENTS

Printed in U.S.A., June 1995
D418009-9741 revision *